Temat pracy :Aplikacja dla Pizzeri

Grupa ………………………….

Bartosz Adamek  
Nr Albumu: 13586

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Wersja | Treść zmiany | Miejsce zmiany | Wprowadził | Obowiązuje od |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| 1.0 | Określenie Smart | Definicja celu | M. Bobrowski | 07.-03-15 |

Tabela wersji dokumentu

SPIS TREŚCI



Aplikacja Pizzerii,

ZNAK GRAFICZNY

CZY SMART CZY CLEAR!!!!??????????

Cele:

1. Możliwość zamawiania pizzy bez stania w kolejce przez aplikacje mobilną lub w sklepie.

2. Zbieranie statystyk na temat zamawianych przez klientów pizzy.

3. Zbieranie ocen pizzy, co pozwala na dopracowanie produktu.

4. Możliwość prostego przydzielania kodów rabatowych.

5. Opcja tworzenia własnej pizzy z dobranymi przez klienta składnikami.

6. Tworzenie rankingów najpopularniejszych pizz stworzonych przez klientów.

7. Możliwość stworzenia konta przez klienta

---------------------

Wymagania funkcjonalnych z atrybutami:

1. Aplikacja posiada minimalną odporność na błędy
   1. Identyfikator: F-01.
   2. Opis: Aplikacja działa bezbłędnie bez potrzeby nadzoru ze strony zespołu technicznego. Zapewnione poprzez stworzenie testów, które aplikacja musi przejść.
   3. Autor: Product Owner.
   4. Źródło: Autor.
   5. Krytyczność: Jedno z najważniejszych wymagań.
   6. Priorytet: Pierwszorzędny.
   7. Wykonalność: Zawansowana
   8. Ryzyko: Podczas użytkowania aplikacji, użytkownik może spotkać się z błędami aplikacji. Rozwiązaniem i minimalizacją tego aspektu jest budowanie aplikacji w sposób obsługi jak największej ilości błędów.
   9. Typ: Niezawodność.
   10. Status: Zaakceptowane.
   11. Planowana iteracja realizacji: 2024
   12. Poziom skomplikowania: Wysoki poziom skomplikowania
   13. Koszt realizacji: 3 miesiące pracy zespołu developerskiego
   14. Właściciel: Product Owner
2. Aplikacja posiada wersje mobilną
   1. Identyfikator: F-02.
   2. Opis: Aplikacja posiada wersję mobilną.
   3. Autor: Product Owner.
   4. Źródło: Autor.
   5. Krytyczność: Jedno z najważniejszych wymagań.
   6. Priorytet: Pierwszorzędny.
   7. Wykonalność: Zawansowana.
   8. Ryzyko: Niekompatybilność ze starszymi wersjami systemów mobilnych. Rozwiązaniem jest upewnienie się, że aplikacja ma niskie wymagania sprzętowe.
   9. Typ: Niezawodność.
   10. Status: Zaakceptowane.
   11. Planowana iteracja realizacji: 2024
   12. Poziom skomplikowania: Wysoki poziom skomplikowania. Zależność od API.
   13. Koszt realizacji: 3 miesiące pracy zespołu developerskiego
   14. Właściciel: Product Owner
3. Aplikacja jest czytelna w użytku
   1. Identyfikator: F-03.
   2. Opis: Aplikacja posiada czytelną czcionkę i kolorystykę.
   3. Autor: Product Owner.
   4. Źródło: Autor.
   5. Krytyczność: Jedno z najważniejszych wymagań.
   6. Priorytet: Pierwszorzędny.
   7. Wykonalność: Zawansowana.
   8. Ryzyko:
   9. Typ: Czytelność .
   10. Status: Zaakceptowane.
   11. Planowana iteracja realizacji: 2024
   12. Poziom skomplikowania: Niski poziom skomplikowania.
   13. Koszt realizacji: Wliczone w koszty tworzenia aplikacji.
   14. Właściciel: Product Owner

4. Aplikacja pozwala na tworzenie niestandardowych pizz przez użytkowników

5. Tworzenie kont użytkowników

6. Zapamiętywania historii kupionych produktów na kontach

7. Możliwość dodawania pizz do listy ulubionych produktów

8. Sugestie produktów dla klientów

9. System lokalnie musi działać bez potrzeby logowania

10. Aplikacja obsługuje popularne systemy płatności

11. Aplikacja informuje o promocjach

12. Aplikacja informuje o dostępności produktu

13. Aplikacja pozwala na wyszukiwanie pizz po składnikach

14. Aplikacja informuje o możliwościach dostawy

15. Aplikacja posiada niskie wymagania, co pozwala uruchomić ją nawet na starych urządzeniach

5 wymagań niefunkcjonalnych z atrybutami:

1. Aplikacja jest zabezpieczona przed atakimi hackerskimi

-Rozwiąanie: Zapewnienie weryfikacji i szyfrowania

-Uzasadnienie: ataki hackerski mogą narazic firme na koszty poprzez powpodowanie awarii systemu

2. Serwer

-Rozwiąanie: implementcja serwera Linuxowego

-Uzasadnienie: serwer jest potrzebny do odbierania zamówien

3. Baza danych

-Rozwiąanie: Implementacja bazy danych na serwerze

-Uzasadnienie: Potrzeba przechowywania danych kont, niestandardowych pizz, opini i statystyk

4. Terminale do zamówień

-Rowiązanie: Implementacja terminali do obsługi zamówień

-Uzasadnienie: Klienci potrzebują posiadać możliwośc zamawiania produktów bez posiadania smartfonów

5. Serwis działa bezbłędnie

-Rozwiązanie: Upewnienie się, że system jest w stanie pracować bez potrzeby ciągłego wsparcia technicznego

-Uzasadnienie: Częsta konserwacja zwiększa koszty użytkowania aplikacji.

To do:   
Określenie celu projektu CO? Cel ma być SMART lub Clear  
Atrybuty do wymagań